In the claims:

The following claim listing replaces all previous claim listings.

1. (Currently Amended) A method operating a computer having a pipelined processor, comprising setting a bit within an instruction test text field of a branch, said bit preventing the branch ~~address~~ addresses from being placed into a branch history table ~~buffer~~ and into a branch target buffer to thereby prevent the branch from being written into the branch history table ~~buffer~~ and branch target buffer and preventing the branch from being predicted and to make the branch only detectable at the time frame of decode.

2. (Original) A method as defined in claim 1 comprising predicting the direction and target of a branch prior to decode.

3. (Original) A method as defined in claim 2 comprising predicting the direction and target of a branch prior to decode through a branch prediction array.

4. (Original) A method as defined in claim 1 comprising tracking the branch from the beginning of the pipe, decode, until the time frame that the given instruction is to be written into a branch prediction array.

5. (Original) A method as defined in claim 1 comprising denoting the instruction text field as a non-writable branch into the BTB.

6. (Previously Amended) A method as defined in claim 5 comprising denoting the instruction field in the system area as a non-writable branch into the BTB in system whereby the branch is blocked from being written to the BTB.

7. (Previously Amended) A method as defined in claim 5 comprising denoting the instruction field in the non-system area, the branch may be predicted via aliasing.

8. (Original) A method as defined in claim 1 wherein machine state altering code lies within an address range spanned by branch tag bits of the branch target buffer.

9. (Canceled)

10. (Original) The method as defined in claim 8 comprising denoting state altering code in the system area by a state bit within the BTB/BHT such that aliasing of branches is prevented within the system area.

Claims 11-30 (Cancelled)